

---

# Harnessing the Informal Processes Around the Computational Design Model

Jonas Runberger and Frans Magnusson

---

## Abstract

This paper presents a strategic framework that facilitates the introduction of computational design techniques into architectural practice. The presented architectural design case, and the strategic framework itself, were developed within the Dsearch, a computational development team part of the R&D at White arkitekter AB. An important aspect of the work within the team is to support the integration of computational processes new to the practice, and promote organisational learning that enables a continuous development. The strategic framework therefore is related to certain concepts within the fields of Sociology and Knowledge Management, such as the notion of boundary objects as first defined by Susan Leigh Star and James R. Griesemer, then later developed by Etienne Wenger as an important factor for collaboration within communities of practice. The strategic framework—an assembly of a number of boundary objects, helps elevate the design model to a design system—a project specific set-up that facilitates design versioning, quality control of processes, and organisational learning. Examples are provided through a case project—the development of a 60 m public bench for Forumtorget (Uppsala, Sweden).

---

## Introduction

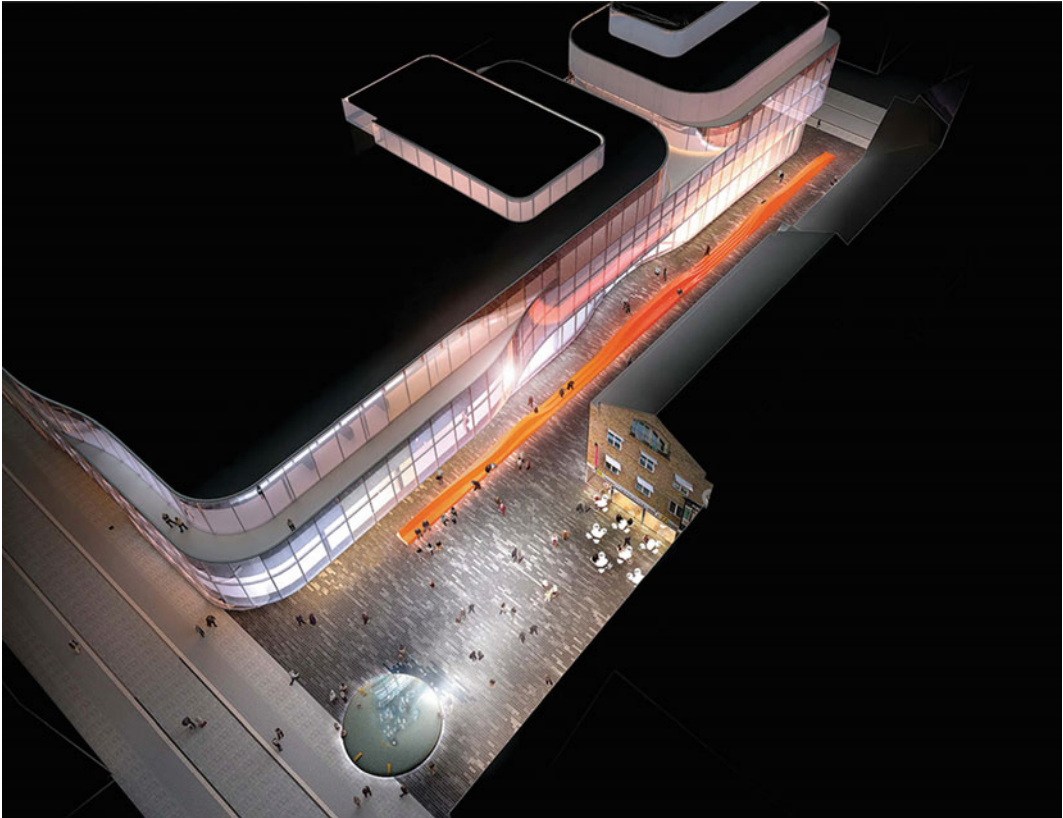
This paper is written from the vantage-point of a team of computational designers, Dsearch, founded with the explicit purpose of introduc-

ing digital and computational strategies and techniques to White arkitekter AB, a large Scandinavian architectural firm. The chosen approach has been to overlay existing projects with computational development, rather than forming a strictly back-office tooling department. The shift from experimental computational practice, to computational design implemented in conventional practice, has highlighted a need for an office infrastructure, mediating between new and old methods and models. The paper details a strategic

---

J. Runberger (✉)  
White arkitekter AB/KTH, Stockholm, Sweden  
e-mail: jonas.runberger@white.se

F. Magnusson  
White arkitekter AB/Chalmers, Gothenburg, Sweden



**Fig. 1** Representation from the competition entry of the Forumtorget case project—the square and urban furniture

framework, informed by current computational design thinking with a theoretical underpinning from the fields of Sociology and Knowledge Management, and presents how it has been deployed by Dsearch to establish this infrastructure. The framework comprises a collection of so-called boundary objects, where the Design System, expanding the scope of a Design Model, plays a vital role (Fig. 1).

## Context

The introduction of computational design into general architectural practice calls for an explicit attention to process and methodology, which in itself can be a provocation to an implicit consensus on methodology—Dsearch has not been preceded by a corresponding “analogue methods

team”. Together with the approach of active project participation by Dsearch in both design and methodological development, within a context having little previous experience of computational design, this may induce an anxious form of organisational learning—one where the stability of architectural practice is shaken.

The most transformative aspect of this change, is the notion of second order development—to borrow the term introduced within cybernetics, i.e. the process of developing a system that in turn generates, regulates and/or evaluates design (Heylighen and Joslyn 2001). Second order development creates models that are associated with first order design processes and procedures—generated automatically or through the interactions with a designer. In a very literal sense design processes are codified into the model alongside actual proposals. The models

produced are at once more powerful, demanding and fragile than traditional design media; they are more open to the receiving context and have embedded agency as long as they are housed within a specific technical and managerial infrastructure.

Second order development also introduces a new specialized culture within architectural practice. Where different participants before could share a drawing, a physical model etc., the design models today also consists of code that requires a new kind of knowledge and technical infrastructure to create, read and use. While graphical programming environments have to some extents lowered the learning curve, there are still distinct boundaries between architects who understand computational design, and those who don't. This is also true for the relation to recently established workflows where BIM software introduces a configuration mode of design. The change in agency of the model media introduced through computational design, together with the cultural and technical implications, often has to be reflected in the mode of operations of second order computational design.

---

## A Strategic Framework of Boundary Objects

Organizational learning—how teams and overall organizations can learn and adapt beyond individual skills and talents, is of great importance to the context of this paper. Most management routines target efficiency and quality in regards to time and delivery, and can thus be related to the idea of single-loop learning—where goals, values and strategies are taken for granted, as opposed to double-loop learning that requires reflection and innovation (Argyris 1999, pp. 68–75). In single-loop learning, each problem is addressed within an existing framework of solutions, with smaller deviations or corrections of errors. Personal experience or already set procedures endure, and a defensive position is established against the unfamiliar. Double-loop learning entails looking at the task at hand with an open mind, and making theories explicit and

clear in order to find new solutions. It is a dialogical, or associative approach—employing new modes of operation to achieve new goals. With this in mind, situating experimental computational design processes in conventional practice requires a strategic framework that allows for double-loop as well as single-loop learning.

Susan Leigh Star and James R. Griesemer's work to understand the communication and cooperation in heterogeneous groups of actors has good bearing on the task set for Dsearch. They developed the concept of *Boundary Objects* to explain the successful cooperation of these actors without explicit consensus regarding the aim of their activities. The basis was the use of common “objects which both inhabit several intersecting social worlds... ..and satisfy the informational requirements of each of them” (Star and Griesemer 1989, p. 393). Flexibility of interpretation combined with their identity and capacity to structure work and flow of information throughout various social groups are key to boundary objects. Star and Griesemer also identify several different types or categories of boundary objects, such as repositories, ideal types (maps or diagrams) or standardized forms.

Etienne Wenger adds another important aspect to this concept; reification—to make the abstract concrete and legible, a process that could be embodied also in artefacts (Wenger 1998, p. 58). Similar to the way that architectural representations act as boundary objects that reify abstract ideas into important steps in a design process (such as sketching and model-making), the reification of boundary objects in a computational design environment may help both internal development and communication to general design teams. This may be even truer for second order design teams with no formal training, i.e. architects shifting from design to computational design.

The way design steps are formalized and made explicit within second order design follows in itself a process of reification, and in this way the computational design model becomes a boundary object that can support collaboration between several specialists. This notion of process as boundary object is a distinct difference to

the conventional practice of end-results (proposals, deliveries) as the boundary objects and clarifies the relation between second and first order of design. In order to facilitate collaboration between first and second order design teams, auxiliary boundary objects such as standards, logs or repository content, are necessary. In this way, the design model can be expanded to a design system—a project specific assembly of boundary objects and actors that supports communication and workflow.

As part of general process development the full set of boundary objects are continuously refined constituents of the strategic framework, in essence a boundary infrastructure (Star 2010, p. 602) that employs specific constellations of boundary objects to establish flows of information and structure work across various environments within White arkitekter, including the internal environment of Dsearch (Fig. 2).

### Deployed Boundary Objects

The *Service Matrix* is a two-dimensional diagram targeting a need to specify and communicate the conditions for and expected benefits of [group] participation in a project to potential collaborators—internal and external to White arkitekter (Fig. 3). Specification is carried out as a continuous dialogue with the project principal to establish an awareness of mutual expectations, as well as a reduction of uncertainty in terms of development time and deliveries. In this sense the matrix fits the description of a boundary object of the Ideal type:

It is abstracted from all domains, and may be fairly vague. However, it is adaptable to a local site precisely because it is fairly vague; it serves as a means of communicating and cooperating symbolically- a ‘good enough’ road map for all parties. (Star and Griesemer 1989, p. 410)

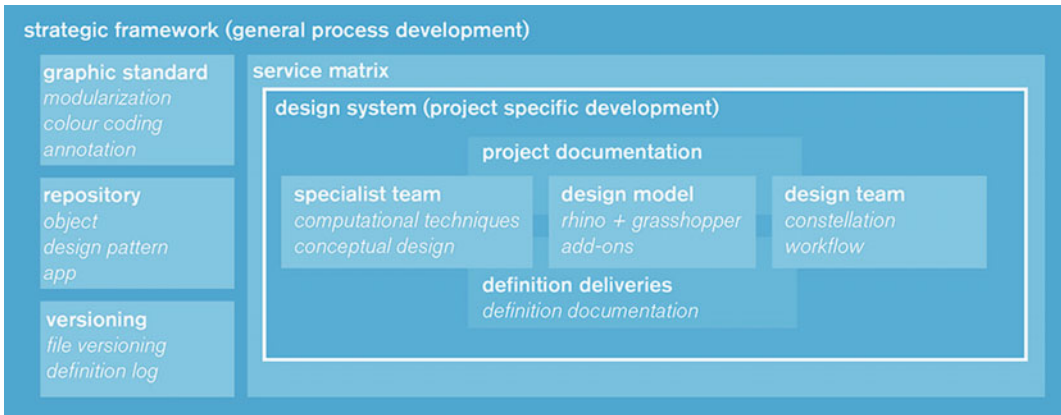


Fig. 2 The strategic framework and deployed boundary objects

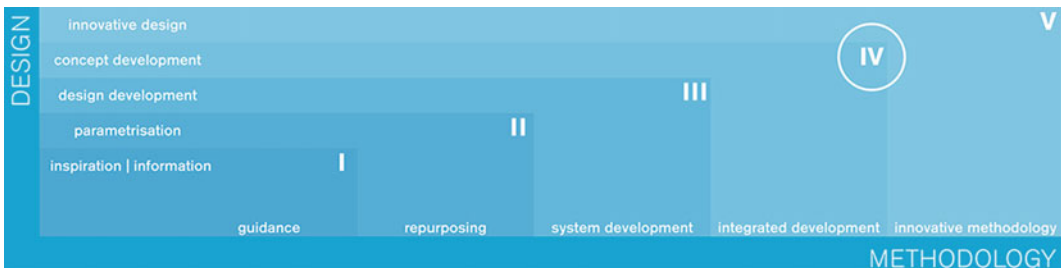


Fig. 3 The service matrix, with the status of the case Forumtorget marked

On the matrix design axis, representing the expected level of design development and architectural innovation, the *Inspiration and Information* level means that the project is discussed in relation to design references and computational methods. In cases where a design concept is well developed and explicitly articulated, a post factum second order implementation is regarded as a *parametrisation*. When Dsearch uses a developed model to deliver a design proposal, it is labelled as *design development* whereas the *concept development* level implies Dsearch taking part in shaping the design concept—introducing computational thinking. *Innovative design* starts with the core value of innovation and is benchmarked against the international design field.

On the methodology axis, representing the level of complexity in computational development, Dsearch can provide *guidance* to the design team regarding relevant methods for the specific project. Design patterns from the repository can be *repurposed* to fit the project needs, but if the necessary modifications are plentiful, complicated or involves new methodology this is considered *system development*. The next level applies when general and computational design methods are intertwined with other specialist methods (energy, daylight, structure etc.) into an *integrated development*. *Innovative methodology* is needed to address problems where no preceding methods are identified.

The *Design Model* is a first and a second order model, coupled; in our case, this currently means a Rhino/Grasshopper, or a Revit/Dynamo model, plus associated data files. When applied in projects, the design model is expanded to a *Design System*, a curated assemblage continuously formed by the models, but also by the design problem at hand and the human actors of the project in addition to methods and tools—bespoke and conventional. The design system is explicitly situated on a specific coordinate in the service matrix. In this way the matrix mediates the formation of each design system, aiding the specification of what resources from the strategic framework and the larger practice will have to be

included in the design system in terms of engagement and expected outcome.

The continuous specification of the design system establishes explicit outer borders for the project—design problem, concept, client intentions, etc.—for both second order development and first order design teams. In that way the strategic framework regulates collaboration and supports the flow of information within and outside of the design system. Star and Griesemer terms this a *Coincident Boundaries Object* where the “...result is that work in different sites and with different perspectives can be conducted autonomously while cooperating parties share a common referent.” (Star and Griesemer 1989, p. 410).

Re-usable sections of definitions are routinely extracted, stored and indexed in one of three formats—Objects, Apps or Design Patterns—at a central *Repository*, along with a documentation of key specifications (Fig. 4). The act of editing and documenting parts for re-use also doubles as an explicit annotation to the original definition. Star and Griesemer mean that the index and modularity of a Repository allows that people “...from different worlds can use or borrow from the ‘pile’ for their own purposes without having directly to negotiate differences in purpose”. One intention behind the repository is just that: hopefully objects developed by Dsearch will find their way to unexpected uses within the larger practice (Star and Griesemer 1989, p. 410). *Objects* (User Objects in Grasshopper nomenclature) wrap a section of the definition into a component with the same affordances as standard components (such as search, legibility and commenting) with the difference that the underlying section still is available for later revision. This format is used for generic processes with an expectedly wide use; different projects or multiple instances within the same definition. *Apps* (in the everyday sense) are complete and freestanding definitions with a specific purpose, as close as Dsearch gets to software development. These repository formats can be said to act as Standardized Form boundary objects in the sense that they strongly regulate the flow of information

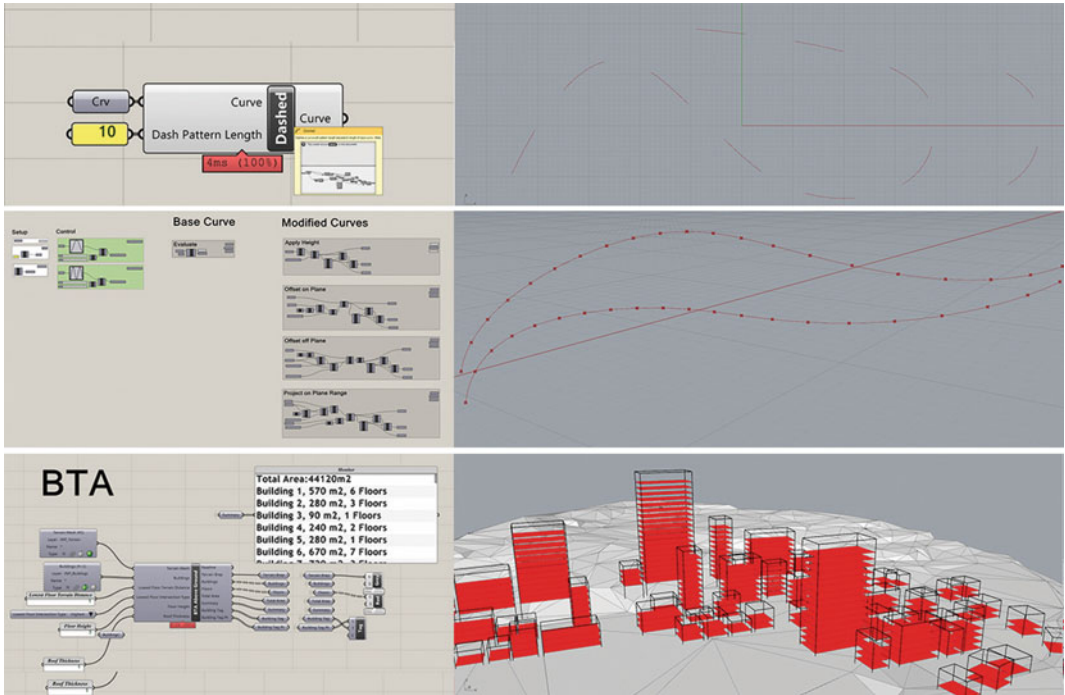


Fig. 4 Examples of objects, design patterns and apps

within a design model in a way that is consistent across all contexts. “The advantages of such objects are that local uncertainties ... are deleted” (Star and Griesemer 1989, p. 411).

*Design Patterns*, “above nodes but below designs” (Woodbury 2010, p. 187), on the other hand, are more complex, specific and often need modification when applied to a new definition context. They may or may not have a distinct

geometrical output, but are seen as distinguishable parts of a definition that provides a particular outcome. Preserving ease of editing is prioritized before application, thus patterns are stored as snippets in the standard definition format.

Other prominent boundary objects of the framework are the *Graphic Standard* (Ideal Type, Fig. 5), stipulating annotation and modularization of the definitions for legibility and

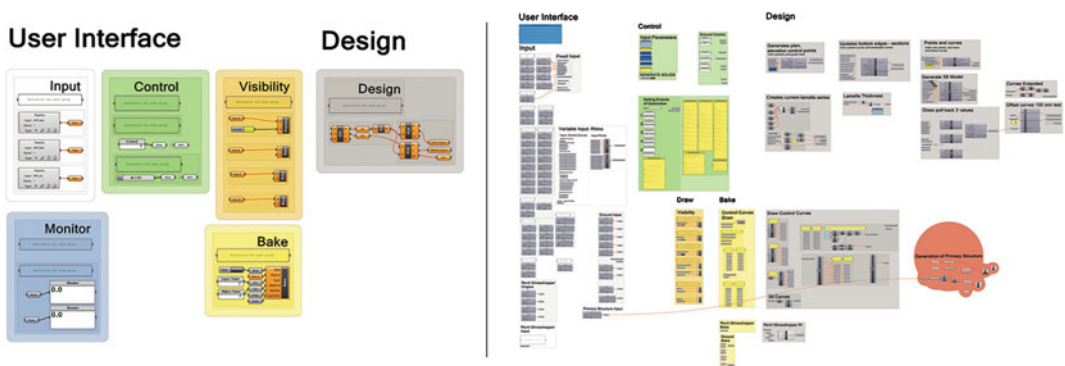


Fig. 5 The graphic standard for grasshopper, as implemented in Forumtorget

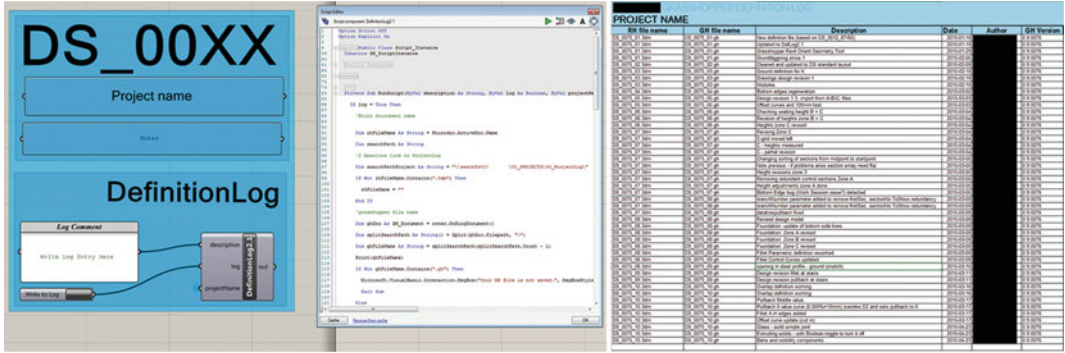


Fig. 6 The definition log object (with script) and definition log

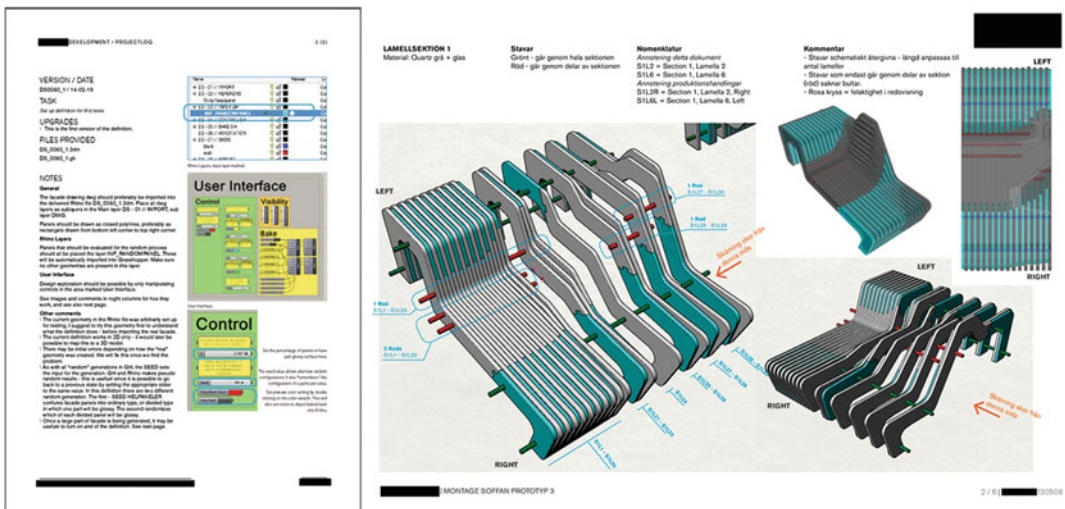


Fig. 7 The definition documentation and project documentation from Forumtorget

reusability (Davis et al. 2011), and different ways of regulating the versioning of design systems—the *Definition Log* object documents brief technical development to a spreadsheet archive, a *Definition Documentation* template is used to communicate changes in functionality between key stages and for deliveries (Fig. 6) and a *Project Documentation* template to communicate design outcome (Fig. 7). As boundary objects they belong to the Standardized Forms type—methods for common communication across dispersed work groups (Star and Griesemer 1989, p. 410). The graphic standard and definition log facilitates easier collaboration within the computational design team, while the definition documentation

documents advanced use and provides relevant information for a general practitioner to use a delivered design system. A separate *Project Documentation* template is used to present design outcome to external parties.

### Applied Strategies in Project Development

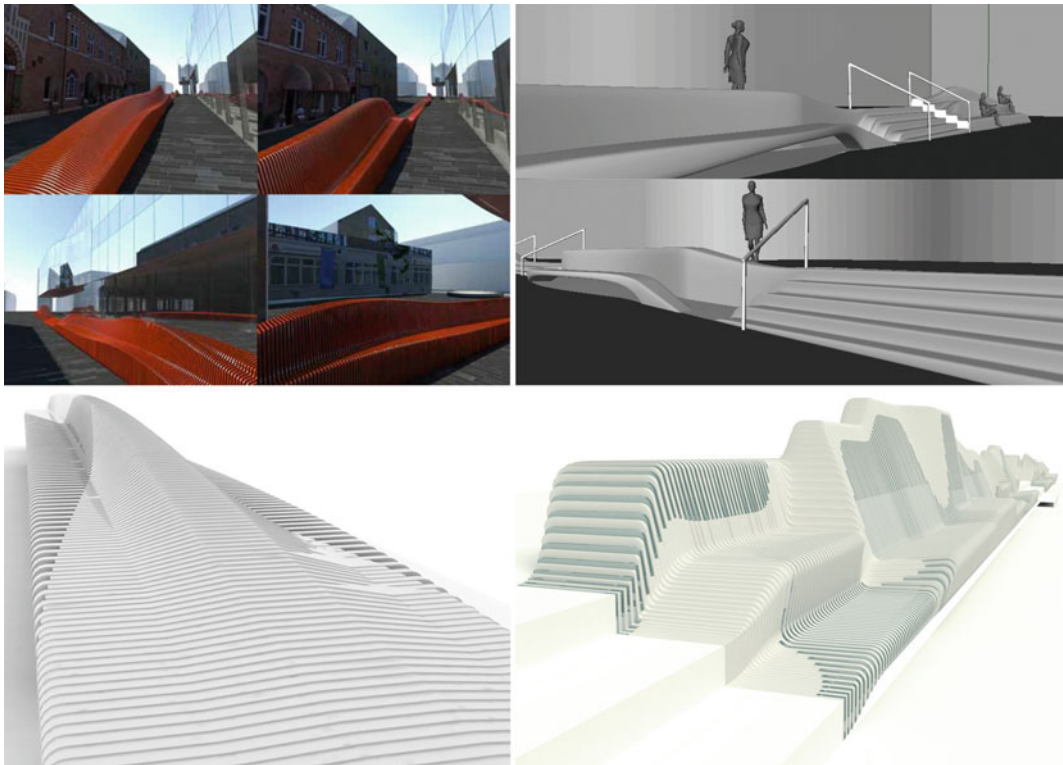
Based on a competition win for Forumtorget, Uppsala, in 2011 (Fig. 1), this urban furniture has been developed through series of computational design models, physical models and prototypes. The programmatic and formal concept includes

the manipulation and variation of seating configurations—the cross section is continuously shifting while providing a variety of seating opportunities on two sides over the 60 m length of the bench. Different ground levels and the integration of stairs into the design added further complexities handled through the design system, which was also set up for deliveries of production documentation. Due to the development and construction of an adjacent building, the design process was extended over time with intermittent activity. This allowed a development in distinct phases, dependent on the strategic framework to make sure that past design evaluations are not lost, and that changes in the development team over time does not impede on development.

The design system enabled that final decisions on form could be postponed to a very late stage, while bespoke technical solutions and general principles were resolved to the level that procurement could be handled. This follows the

arguments made by Daniel Davis as he challenges the conventional model of high influence and low expenditure in early project phases, claiming that parametric models can “shift the cost of change in relation to design effort”, “allowing designers to defer key decisions until later in the project” (Davis 2013, p. 208). In the experience of the authors, this is indeed possible, but in relation to overall project workflows it has required the strategic framework that expands the design model to a design system, and provides boundary objects that supports interactions over time.

The Forumtorget project has undergone four main development stages (Fig. 8). In the first competition stage, a simple design presented variation as a concept, developed through a basic computational model with visual representations as the main outcome. In the second stage, form development was conducted in order to set the boundaries in relation to design identity and site



**Fig. 8** Representations of Forumtorget from stages 1, 2, 3 and 4



specific conditions such as the two different levels of ground. The third design stage involved an iterative study of the form, structure, manufacture and assembly of the bench, through a continuously refined parametric model, with input from specialists as well as feedback from series of physical models and full-scale prototypes.

During the fourth stage, the technical details and structure were refined, the overall form revised, the design system consolidated to avoid errors in final production, and the generation of documentation for procurement and production initiated. From stage two to four, care was taken to keep exploring the overall spatial principles, yet not fixating any part completely.

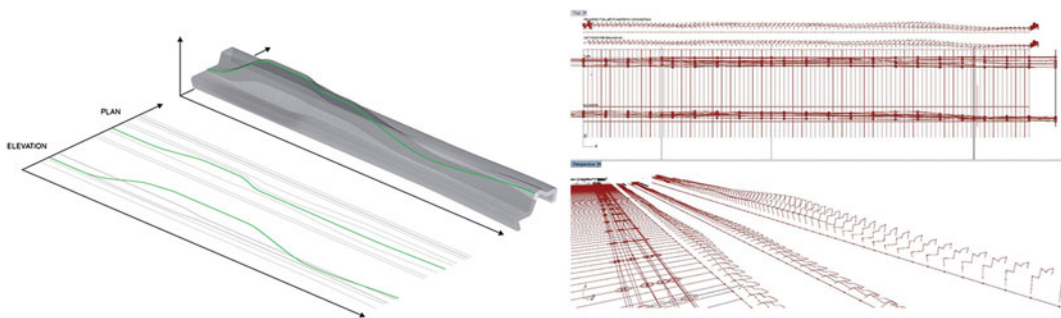
A number of early key decisions allowed the continuous development based on the design system. A lamella-based form and fabrication principle that allowed rational fabrication of the “free-form concept” was selected. The cross-section was based on a seven segment polyline with individual control of all corners and their fillets, setting restrictions of possible formal variation, yet handling the zone-specific conditions and providing a formal continuity along the length of the bench as well as an adaptation to differentiated ground conditions. The use of control diagrams; the computational transformation of series of 2d curves to the design model, provided a control interface to overall form as well as detailing (Fig. 9).

Once the premises for the design system were in place, the project was developed through the established versioning approach facilitated

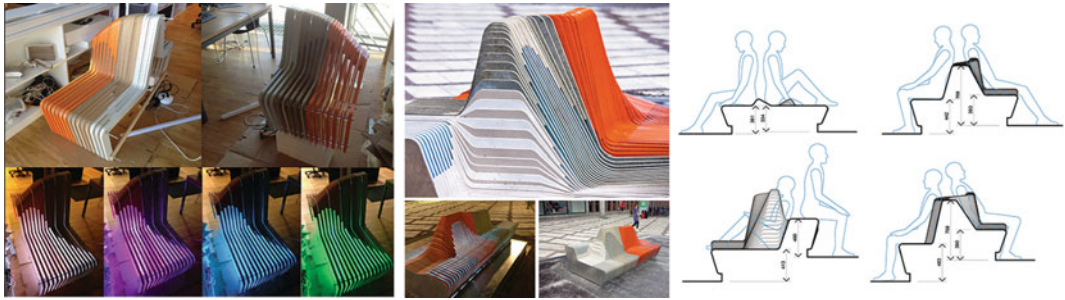
through the strategic framework. Supported by a series of design reviews where alternate solutions could be discussed with different specialists, the second order development could operate through iterative loops, allowing technical solutions to be set while the specific form was continuously refined. The use of the definition log also provides a back-log of the development, where the following milestones can be identified:

- Versions 5–8: Setting up basic geometrical generation from control curves (Fig. 9).
- Versions 23: Generating production documentation for first prototype (Fig. 10).
- Version 38: Optimizing definition performance by introducing python scripts.
- Versions 41–58: Developing second prototype (Fig. 10).
- Version 88: Setting up relation to Revit for interface to overall project.
- Version 90: Setting up modularization for production and assembly.
- Version 95: Developing concrete foundations design model.
- Version 97: Developing joint between glass and solid materials.
- Version 100: Overall formal design revisions.
- Version 101: Developing primary steel structure.
- Version 102: Generating procurement documentation (Fig. 11).

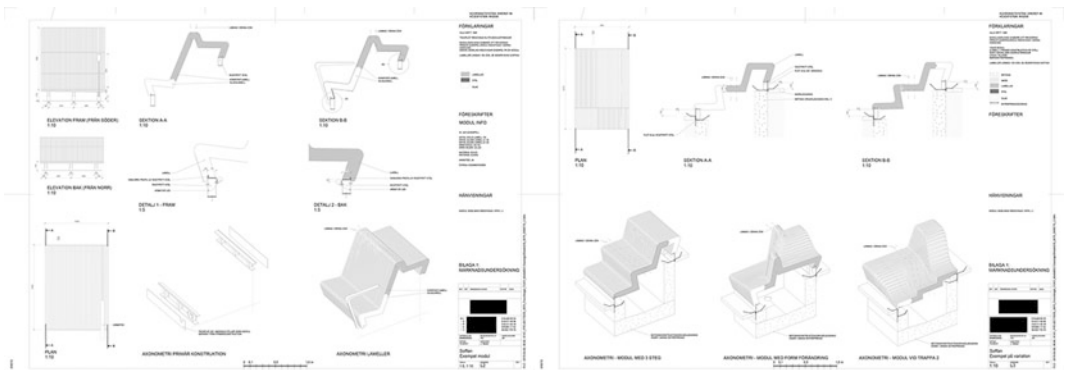
The extended development time of the project made documentation very important—a number of different Dsearch members have entered and



**Fig. 9** Control curves representing plan and elevation as a diagram, and view from the design model



**Fig. 10** First and second physical prototype, and representations from stage 3



**Fig. 11** Procurement documentation from stage 4

exited the project, and the definition log and definition documentation allowed an understanding of previous development. The close collaboration with other specialists and the client also required a continuous use of a project documentation, in particular the development and assembly of full scale prototypes. The overall design process in this way reflected an exploration of two related trajectories; the design development based on several specialisms, where issues such as comfort, identity, materials, structure and production informed the ongoing versioned computational development. In the fourth stage all these issues converged into a close to final design proposal, where also computational principles for procurement and production documents were part of the design system.

## Concluding Remarks

Computational and second order design provides architectural practice with new assets, but the deployment in practice may face unprecedented challenges. This paper shows how a strategic framework can be applied in order to establish an infrastructure that in practice mitigates these challenges. Together with general design management, it affects design process and outcome, as well as organizational learning. In relation to this paper, single- and double-loop learning in computational design can be regarded in two opposing ways; one the one hand, computational design is highly procedural—with the proposed strategic framework that may seem even more

regulated—suggesting single-loop learning within a given framework; on the other hand, it can be regarded in contrast to existing management models within architectural practice, where the procedures taken are very familiar—we know the modes of representation, (we believe) we know what a client wants, and we understand how the construction industry operates. Given that computational design provides new opportunities in conventional practice, but depend on special skills and a rapid technological advancement—the purpose is to use the strategic framework to enable a continuous exchange between first and second order design, providing double loop learning within both contexts. As the field of computational design continues to develop, the framework enables additional aspects to be introduced, such as iterative analysis, material performance and additive manufacturing.

The strategic framework and its constituent bounding objects is to be seen as a bottom-up approach to managing design processes and flows of information across a heterogeneous environment within a larger firm. Grounded in computational design thinking, it provides an alternative to current trends of explicit process modelling and comprehensive BIM standardization. The associated terminology from Sociology and Knowledge Management has in the development of Dsearch provided useful concepts bridging between first and second order of design and addressing issues beyond computation and design—the interactions between designers of different backgrounds. In essence, it provides a deeper understanding of elements that previously made sense in an intuitive way and articulates

them. The terminology opens up these interactions for a wider discussion—by aligning ourselves with recent research on practice cultures elsewhere, we are able to distinguish and evaluate potential assets for future practice and research.

**Acknowledgements** Forumtorget design team: Gustav Jarlöv, Sam Keshavarz, Jens Modin, Andreas Milsta, Torbjörn Eliasson, Michael Malmborg, Jonas Wannfors. Dsearch team: Jonas Runberger, Frans Magnusson, Hamia Aghaiemeybodi, Pedram Seddighzadeh Yazdi, Vladimir Ondejcik. All images © White arkitekter AB.

---

## References

- Argyris C (1999) *On organizational learning*, 2nd edn. Blackwell, London
- Davis D (2013) *Modelled on software engineering: flexible parametric models in the practice of architecture*. PhD dissertation. RMIT University, Melbourne
- Davis D, Burry J, Burry M (2011) Understanding visual scripts: improving collaboration through modular programming. *Int J Architectural Comput* 9(4):361–376
- Heylighen F, Joslyn C (2001) Cybernetics and second-order cybernetics. In: Meyers RA (ed) *Encyclopedia of physical science and technology*, 3rd edn. Academic Press, New York
- Star SL (2010) This is not a boundary object: reflections on the origin of a concept. *Sci Technol Hum Values* 35:601
- Star SL, Griesemer JR (1989) Institutional ecology, translations' and boundary objects: amateurs and professionals in Berkeley's Museum of Vertebrate Zoology 1907–39. *Soc Stud Sci* 19(3):387–420
- Wenger E (1998) *Communities of practice, learning meaning and identity*. Cambridge University Press, Cambridge
- Woodbury R (2010) *Elements of parametric design*. Routledge, London